Применение методов оптимизации и машинного обучения для оценивания шанса поступления абитуриентов в ВУЗ

Рябинин Евгений
Optimization Class Project. MIPT

Введение

Оценивание вероятности поступления определенного абитуриента в высшие учебные заведения по различным показателям вступительных данных является актуальной задачей для большинства ВУЗов. Чтобы изучить эту проблему качественнее, я решил воспользоваться открытыми данными о поступлении 400 абитуриентов в один из международных институтов.

Данные

Данные были взяты из открытого источника на сайте kaggle.com. Датасет представляет из себя информацию о 400 абитуриентах со следующими параметрами:

0. Serial No. (out of 400)
1. GRE Scores (out of 340)
2. TOEFL Scores (out of 120)
3. University Rating (out of 5)
4. Statement of Purpose (out of 5)
5. Letter of Recommendation Strength (out of 5)
6. Undergraduate GPA (out of 10)
7. Research Experience (either 0 or 1)
8. Chance of Admit (ranging from 0 to 1)

0 параметр является техническим и не используется в дальнейшем. 1-7 являются критериями поступления , по которым оценивался шанс поступления. 8 параметр отвечает за шанс поступления в институт. Пример данных можно видеть на рис. 1. Краткие характеристики указаны на рис. 2. Подробнее с данными можно ознакомиться здесь.

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

(a) Рис. 1

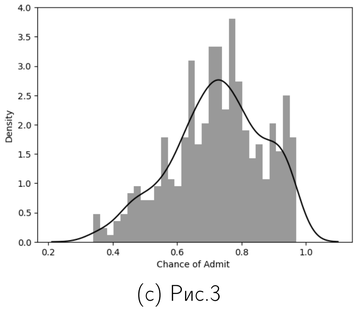
	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	200.500000	316.807500	107.410000	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	115.614301	11.473646	6.069514	1.143728	1.006869	0.898478	0.596317	0.498362	0.142609
min	1.000000	290.000000	92.000000	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	100.750000	308.000000	103.000000	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000
50%	200.500000	317.000000	107.000000	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	300.250000	325.000000	112.000000	4.000000	4.000000	4.000000	9.062500	1.000000	0.830000
max	400.000000	340.000000	120.000000	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000

(b) Рис.2

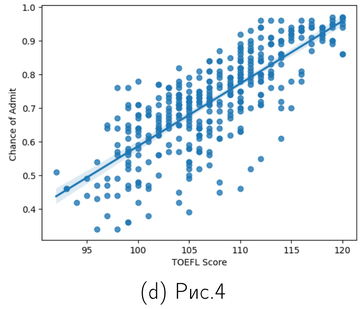
Датасет используется для построения линейной регрессионной модели различными методами. Ошибки моделей сравниваются с текущими показателем шанса поступления конкретного абитуриента из таблицы посредством метрик MSE и MAE.

Препроцессинг

В Google Colab на языке Python был проведен препроцессинг наших данных. Посчитаны статистические данные по каждому из критериев и построены двумерные графики зависимостей между всеми параметрами. Пример результатов продемонстрированы на рисунках. Подробную статистику можно посмотреть по следующей ссылке.



(c) Рис.3



(d) Рис.4

Постановка задачи

В задаче наименьших квадратов, или линейной регрессии, у нас есть измерения $X \in \mathbb{R}^{m \times n}$, $y \in \mathbb{R}^m$ и ищем вектор $\theta \in \mathbb{R}^n$ такой что $X\theta$ приближен к y. Слизость определяется как сумма квадратов разностей:

$$\sum_{i=1}^{m} \left(x_i^\top \theta - y_i \right)^2$$

также известен как l_2-норма квадрата, $\|X\theta - y\|_2^2$

У нас есть датасет из $m = 400$ абитуриентов, каждый из которых представлен $n = 7$ признаками. Каждая строка $x_i^\top$ из X это функции для абитуриента i, в то время как соответствующая запись y_i из y это измерение, которое мы хотим предсказать на основе $x_i^\top$, в данном случае шанс поступления. Предсказание дается с помощью $x_i^\top \theta$.

Мы находим оптимальный θ, решая задачу оптимизации

$$\|X\theta - y\|_2^2 \to \min_{\theta \in \mathbb{R}^n}$$

Пусть θ^* обозначает оптимальный θ. Величина $r = X\theta^* - y$ известна как отклонение. Если $\|r\|_2 = 0$, мы имеем идеальное приближение.

Методы решения

Задача линейной регрессии решалась тремя методами:

1. Многомерная линейная регрессия. Метод реализован на языке Python посредством библиотеки scikit-learn.
2. CatBoost. Метод реализован на языке Python посредством класса CatBoostRegressor
3. Решающие деревья. Метод был реализован ная языке R посредством открытого источника.

Построение первых двух моделей можно посмотреть в конце этого файла. Материалы по третьему методу находятся в этой папке.

Метрики качества

1. MSE (средняя квадратичная ошибка). Чтобы рассчитать MSE, вы берете разницу между предсказаниями вашей модели и основополагающей правдой, возводите ее в квадрат и усредняете ее по всему набору данных.

$$MSE\left(y^{\text{truse}}, y^{\text{pred}}\right) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - f\left(x_i\right)\right)^2$$

2. MAE (средняя абсолютная ошибка). Чтобы рассчитать MAE, вы берете разницу между предсказаниями вашей модели и основополагающей правдой, применяете абсолютное значение к этой разнице и затем усредняете его по всему набору данных.

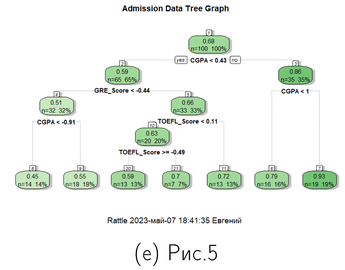
$$MAE\left(y^{\text{true}}, y^{\text{pred}}\right) = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - f\left(x_i\right)\right|$$

Результаты

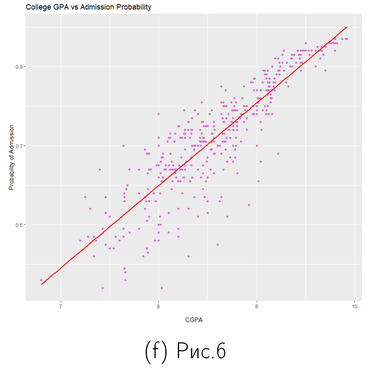
В результате были получены различные диаграммы для каждого критерия и двумерные графики зависимостей всех вариаций двух параметров датасета в рамках препроцессинга. После тремя методами решена задача линейной регрессии, результаты получены в виде списков, графиков, деревьев и метрик качества.

На рис.5 представлен пример деревьев решений нашей задачи, реализованное третьим методом. На рис. 6 представлен график зависимости шанса поступления от CGPA методом МНК. Все результаты находятся на Github

Результаты реализованных методов кореллируют с исходными данными.



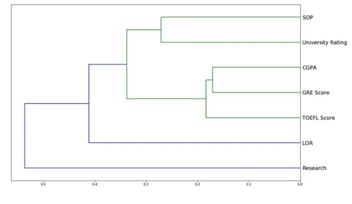
(e) Рис.5



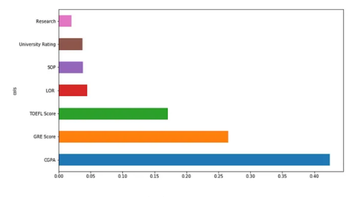
(f) Рис.6

Заключение

По результатам работы оценена важность каждого признаков нашей модели, отвечающей за шанс поступления, т.е. от чего в наибольшей степени зависит вероятность поступления и как сильно взаимосвязаны различные параметры. В открытом источнике преоставлены подобные данные. На рис.7 представлена дендограмма признаков от вероятности поступления, которая показывает взаимосвязь каждого из параметров. Чем раньше пересекаются объекты на рисунке (если смотреть с правой стороны), тем сильнее они взаимосвязаны. Более упрощенный вид зависимости продемонстрирован на рис.8 в виде диаграммы.



(g) Рис.7



(h) Рис.8

Источники

[1] - исходные данные для реализаии методов и препроцессинга
[2] - коллаб с реализацией методов и препроцессинга на Python
[3] - исходный Github с материалами
[4] - сайт fmin.xyz, раздел линейной регрессии
[5] - сайт machinelearning.ru с необходимыми материалами по теории машинного обучения
[6] - открытый источник, необходимый для реализации метода решающих деревьев на языке R
[7] - подобное исследование шанса поступления по таким же данным